A Novel Active Disturbance Rejection Control of PMSM Based on Deep Reinforcement Learning for More Electric Aircraft

Yicheng Wang, Shuhua Fang, *Senior Member, IEEE*, Jianxiong Hu, *Student Member, IEEE*, and Demin Huang

***Abstract*—In this article, a novel active disturbance rejection control (ADRC) based on deep reinforcement learning (DRL) is proposed to improve the performance of permanent magnet synchronous motor (PMSM) for more electric aircraft (MEA). MEA motors have the requirements of safety and stability so that a new ADRC method is put forward based on the limitation of nonlinear error attenuation function of traditional ADRC. The flux weakening control model of PMSM is firstly established. Then the ADRC model is built and applied to the speed loop of the control system. In order to reduce the number of control parameters and the jitter of the control law, deep neural network is employed to replace the traditional control law. Markov decision process is integrated into the novel ADRC to establish DRL model. A method based on twin delayed deep deterministic (TD3) policy gradient algorithm is proposed to train the neural network and optimize the DRL model. Model predictive control and traditional ADRC are used as comparison algorithms. Simulation and experiments show the effectiveness and superiority of the proposed method.**

***Index Terms*—Active disturbance rejection control; deep reinforcement learning; twin delayed deep deterministic; more electric aircraft; permanent magnet synchronous motor; flux weakening.**

I. Introduction

MORE electric aircraft (MEA) has attracted more attention due to the rising demand for more air transportation and dealing with anticipated economic and environmental impacts[1]. Permanent magnet synchronous motor (PMSM) is employed for MEA according to its simple structure and high power density [3][4]. Flux weakening (FW) control is often applied to obtain a wide speed range for the requirement of MEA motor [5]. Feedforward FW control can generate larger torque in high speed region and provide wider constant power region, but has the disadvantage that relying on the accuracy of parameters [6]-[8], which implies that the feedforward FW method is sensitive to the disturbance of internal parameters of the motor. External disturbance represented by wind for MEA motor is also another influence factor which should be considered when the MEA motor control system is built [9][10].

It is an advanced practice to combine the controller method to improve the control effect and enhance the disturbance rejection ability of the system. Model predictive control (MPC) is a popular method for PMSM control, which can improve the response speed of the control system and obtain better dynamic performance [11]. Sliding mode control (SMC) is also employed as adaptive controller in the PMSM system to achieve fast convergence rate of the tracking error [12]. However, SMC usually requires the large control gain and may exhibit chattering behavior [13]. MPC has provided a faster dynamic performance and has good ability for disturbance rejection, but it is sensitive to parameter variations [14].

Active disturbance rejection control (ADRC) which has simple structure and strong robustness was proposed in [15]. ADRC has suppression effect on both internal and external disturbance [16], which is suitable to overcome the shortcoming of feedforward FW. Many researchers have combined ADRC with other control methods to achieve better control effect. Chen proposed a linear ADRC which was combined with model compensation and can reduce the rotor displacement and torque ripples [17]. In [18], an adaptive ADRC with closed-loop flux observer was proposed to solve the noise sensitivity for speed-sensorless induction motor drives. Combining ADRC with other methods can improve the overall control effect, but it doesn't fundamentally improve ADRC itself. The characteristic of nonlinear error decay function of ADRC was analyzed, which could be limited under complex working conditions [19].

The components and structure of ADRC were also researched to improve the performance of the controller. A novel nonlinear function was proposed to replace the traditional *fal*() function of ADRC and further expanded the disturbance range so as to improve the robustness [19]. An improved error feedback control law and a sliding mode controller based on an improved exponential convergence law were proposed to replace the control law link and improve the dynamics performance of ADRC, respectively [20][21]. The improvement of the controller and control law enhances the disturbance rejection performance of the motor effectively, but the motor operation is complex, and different error attenuation functions have their own scope of application [22], which may not meet the

This paragraph of the first footnote will contain the date on which you submitted your paper for review, which is populated by IEEE.

This work was supported in part by the Aeronautical Science Foundation of China under Grant 20160769002 and in part by the Six Talent Peaks Project of Jiangsu Province under Grant GDZB-103. *(Corresponding author: Shuhua Fang)*

The authors are with the School of Electrical Engineering, Southeast, University, Nanjing 210096, China (email: 230218324@seu.edu.cn; shfang@seu.edu.cn; 220182640@seu.edu.cn; 220202916@seu.edu.cn)

requirements of different situations, which means the control law itself has limitations.

Based on the current development of computer technology, machine learning can use experience and exploration to build the model of unknown systems [23]. As a kind of machine learning, deep reinforcement learning (DRL) has been successfully applied in other fields. In [24], a decision-making scheme based on DRL was proposed for dual active bridge converter to give the optimal control decision to minimize the power loss. DRL was also used to observe the output feedback during Li-ion battery charging to obtain the shortest charging time without damaging the battery [25]. A novel method was proposed to summary the relationship between the control effect with parameters, and obtain the optimal parameters of PI controller of PMSM [26]. As a kind of unsupervised learning method, DRL can use the maximum likelihood method to establish a direct relationship between the observations and the required outputs, explore and update the optimal relationship by strategies, reward function, and continuous interaction with the environment, so as to achieve the same effect as the control law [27], or even better.

In this article, a novel ADRC based on DRL is proposed for MEA PMSM control. The innovative contributions of this article are: 1) An innovative ADRC framework is proposed. This article combines DRL with ADRC and proposes a novel TD3-ADRC based on DRL. The limitation of the existing control law is solved by taking the deep neural network and training by TD3 algorithm in DRL. 2) Combining DRL with motor control system, a TD3-ADRC training method is proposed. The interface module and reward function of DRL in ADRC and motor control system are designed. Based on ADRC control system, Markov decision process is established to guide the interaction between Agent, motor model and interface module. The control network is trained autonomously by employing TD3 algorithm. Specifically, the mathematical model of ADRC which is employed as speed loop controller is built based on feedforward FW system in Section II. ADRC control theory and Markov decision process (MDP) are combined, DRL model and interactive interface applied in ADRC are established in Section III(A)(B). Twin delayed deep deterministic policy gradient algorithm (TD3) is employed as policy search algorithm in Section III(C). The novel TD3-ADRC control system is simulated, and experiments are carried out to verify the effectiveness and progressiveness of the proposed method in Sections IV and V.

II. Design of the ADRC

Traditional ADRC consists of tracking differentiator (TD), extended state observer (ESO), and nonlinear state error feedback control law (NLSEF). Fig. 1 shows the control framework of ADRC.

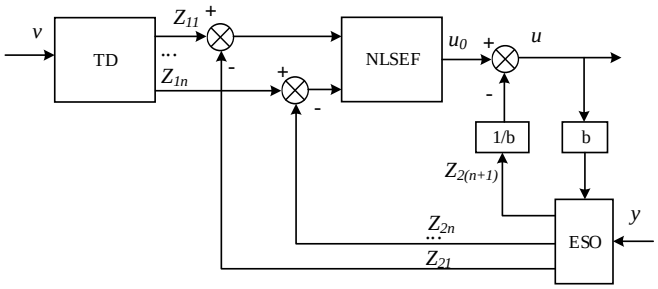


Fig. 1. Control framework of ADRC.

In the control framework of ADRC, v is the reference input. Through the arrangement of the transition procedure, TD outputs derivative signal $Z_{11},\ldots,Z_n$. The extended state $Z_{21}...Z_{2(n+1)}$ are obtained when the output y of the whole system is analyzed by ESO. The compensation coefficient is set as $1/b$, and the compensation is obtained by the total disturbance $Z_{2(n+1)}$.

In order to make up for the shortcomings of FW and play a better control role, ADRC is applied in the one-order speed loop of the control system, and the mathematical model of ADRC is built as follows.

A. Tracking Differentiator

$$TD\begin{cases} e_1 = z_{11} - \omega^* \\ \dot{z}_{11} = -r fal(e_1, a_1, \delta_1) \end{cases} \tag{1}$$

$$fal(x,a,\delta) = \begin{cases} \dfrac{x}{\delta^{1-a}}, |x| \le \delta \\ sign(x), |x| > \delta \end{cases} \tag{2}$$

where r is the constant coefficient. ω^* is the difference between the given speed and the actual speed. The extended state $fal()$ is a non-linear function, $sign(x)$ is a symbolic function, when $x \geqslant 0$, $sign$=1, and when x<0, $sign$=0, a is the nonlinear factor, and δ is the filter factor.

ADRC uses TD to track the reference input of the system and arranges the transition process.

B. Extended State Observer

$$ESO\begin{cases} e_2 = z_{21} - \omega \\ \dot{z}_{11} = z_{22} - \beta_{21} fal(e_2, a_{21}, \delta_{21}) + bu \\ \dot{z}_{22} = -\beta_{22} fal(e_2, a_{22}, \delta_{22}) \end{cases} \tag{3}$$

where ω is the observed speed used for feedback, u is the control effect, that is, the output of the system, and β_{21}, β_{22} are constants.

ADRC refers to the disturbance caused by the parameters that change with time in the motor as internal disturbance. ESO observes and estimates the total disturbance composed of the internal disturbance and the external disturbance caused by external changes, and analyzes the output and differential state of the system.

C. Nonlinear State Error Feedback

$$NLSEF\begin{cases} e = z_{21} - z_{22} \\ u_0 = \beta_3 fal(e, a_3, \delta_3) \\ u = u_0 - z_{22}/b \end{cases} \tag{4}$$

where β_3 is constant.

NLSEF analyzes the output states of TD and ESO, and adjusts the output errors of TD and ESO by using nonlinear control law to achieve stable control effect.

D. Total design of ADRC for FW

Fig. 2 shows the control system combining ADRC and feedforward FW. Due to the output of ADRC is T_e, compensation coefficient b is obtained according to the motion formula

$$\frac{d\omega}{dt} = \frac{1}{J}T_e - \frac{1}{J}T_L - \frac{1}{J}B\omega_m \tag{5}$$

Owing to the output u of the speed loop is bT_e, b can be calculated as $1/J$.

It can be seen from Figs.1 and 2 that NLSEF link is essentially based on the judgment of the relationship between input and output, using or designing corresponding mathematical laws, and giving different outputs according to the input. Therefore, the proposed DRL-method based on experience and exploration can achieve this more comprehensively.

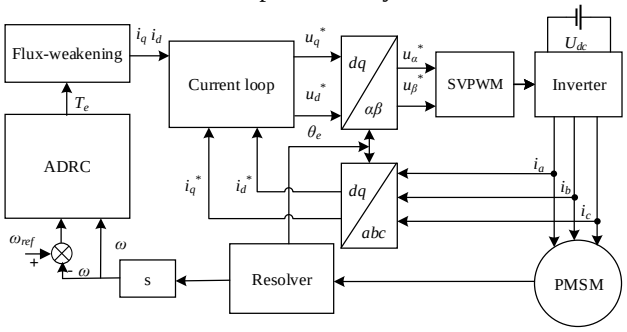


Fig. 2. Structure of FW control system based on ADRC.

III. Framework of TD3-ADRC

A. Framework of ADRC Combined with DRL

Fig. 3 shows the basic principle of employing DRL as input-output algorithm in this article. DRL algorithm trains Agent to explore and learns the relationship between input and output through its interaction with the environment model. *Environment*, *State*, *Action*, and *Reward* are professional terms in DRL, which are distinguished by italicized capital letters. In the current *Environment*, based on the input and *State S*, Agent selects an *Action A* to interact with *Environment*, and obtains the new *State S* generated after the occurrence of *A*, as well as the evaluation *Reward R* for *S*. According to *Reward*, Agent will improve each *Action A*, and constantly interact and evaluate with the environment until *R* converges. This is the MDP combined with input-output observation which constitutes the basis of the DRL optimization model.

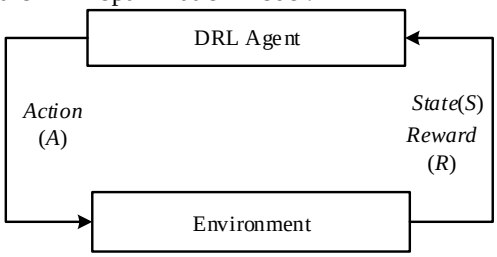


Fig. 3. DRL method for input-output fitting algorithm.

Fig. 4 shows the control logic block diagram of TD3-ADRC which combines ADRC and MDP. TD3-ADRC Agent regards the whole motor control system as *Environment*, the motor speed as *State*, and the motor control effect as *Reward*. TD3-ADRC Agent determines *Action A*, that is, the fitting value of the output based on the current *Environment* and input. Then *Environment* gives a new *State S* according to the output value and calculates *Reward* value *R*. TD3-ADRC Agent makes judgment, optimizes and updates the next *Action*, and interacts with the *Environment* until the *Reward* value converges. The trained Agent can decide how to optimize and select *Action*s by itself. Fig. 5 shows the specific flux-weakening control system based on TD3-ADRC.

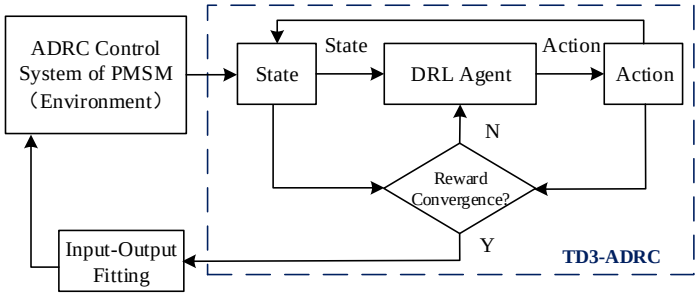


Fig. 4. Control logic of TD3-ADRC.

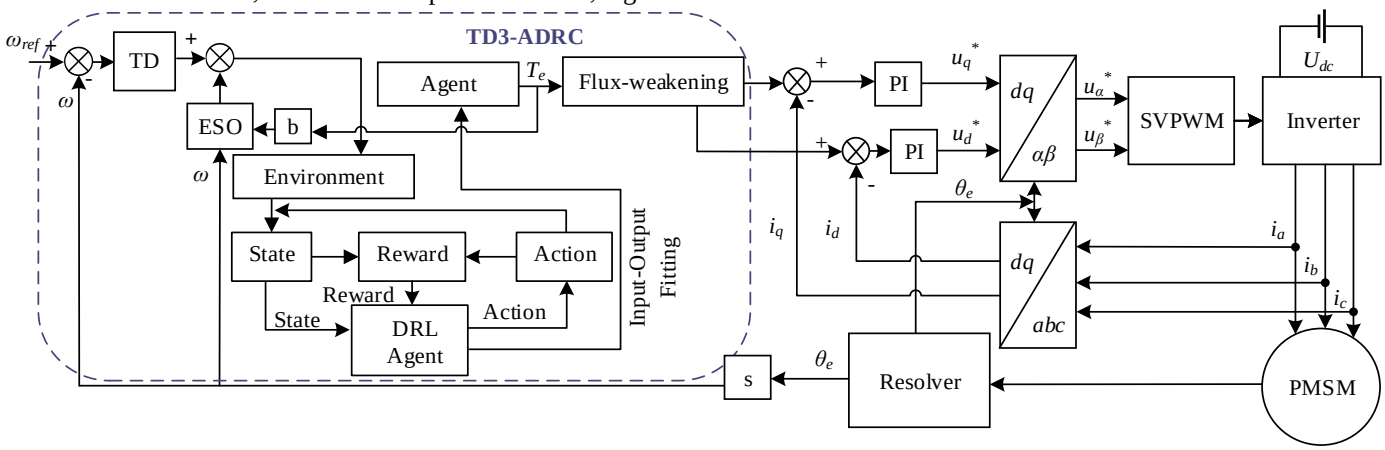


Fig. 5. TD3-ADRC control system based on DRL of PMSM for MEA.

B. DRL Network and Model Establishment

The mapping from *State* s_t to *Action* a_t is called policy π at the current time t

$$a_t = \pi(s_t) \tag{6}$$

Reward value r_t and next State s_{t+1} can be obtained after a_t becomes effective. By repeating MDP process, new states and reward values can be obtained continuously according to actions produced by π. The cumulative Reward value can be calculated as

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} = r_0 \cdot \gamma^0 + r_1 \cdot \gamma^1 + r_2 \cdot \gamma^2 + \ldots \tag{7}$$

where discount coefficient γ is employed to prevent G_t from growing indefinitely in the case of infinite step size.

The purpose of DRL is to find the optimal policy, which can maximize the cumulative *Reward* value. The expected value of the cumulative total return is called the state value function

$$V_\pi(s) = E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s] \tag{8}$$

The Behrman equation of (8) can be written as

$$\begin{aligned} V_\pi(s) &= E_\pi[\sum_{k=0}^{\infty} G_t \mid S_t = s] \\ &= E_\pi[r_{t+1} \cdot \gamma^0 + r_{t+2} \cdot \gamma^1 + \ldots \mid S_t = s] \\ &= E_\pi[r_{t+1} \cdot \gamma^0 + \gamma(r_{t+2} + r_{t+3} \cdot \gamma + \ldots) \mid S_t = s] \\ &= E_\pi[r_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= E_\pi[r_{t+1} + \gamma V_\pi(s_{t+1}) \mid S_t = s] \end{aligned} \tag{9}$$

Corresponding to Eq.(8), considering the effect of *Action* on value function, the state-action value function can be written as

$$\begin{aligned} Q_\pi(s_t, a_t) &= [r_s^a + \gamma V_\pi(s_1)|_{s_1}] \mid S_t = s, A_t = a \\ &= E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s, A_t = a] \end{aligned} \tag{10}$$

The Behrman equation of (10) can be written as

$$Q_\pi(s_t, a_t) = E_\pi[r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1}) \mid S_t = s, A_t = a] \tag{11}$$

Each policy corresponds to a state-action value function, which means that the optimal policy corresponds to the optimal state-action value function. The optimal value function marked by * represents the value function with the largest value in all policies. Therefore, the optimal Behrman equation can be expressed as

$$Q_\pi^*(s_t, a_t) = r_{s_t}^{a_t} + \gamma Q_\pi^*(s_{t+1}, a_{t+1}) \tag{12}$$

The optimal policy π^* can be found according to the maximum cumulative return and its Behrman equation

$$\pi(s) \to a_t = \arg\max Q_\pi^*(s_t, a_t) \tag{13}$$

C. TD3 algorithm based on DRL

One of the methods to solve MDP is policy search, which parameterizes the policy into $\pi(s|\theta^\pi)$, and uses parameterized functions (such as neural networks) to represent the policy and find the optimal policy parameters θ^π. The selection of DRL algorithm in improved ADRC application needs to consider the following situations: 1) The input and output are distributed in the continuous space, and the Actor-Critic structure methods (AC) are algorithms that satisfies the continuous action space. 2) The deterministic strategy algorithm has higher efficiency in the case of less sampling data. Based on these two points, TD3 deterministic strategy algorithm with AC structure, a state-of-the-art deterministic strategy algorithm [28], is selected as the DRL algorithm in this article. The algorithm structure is shown in Fig. 6.

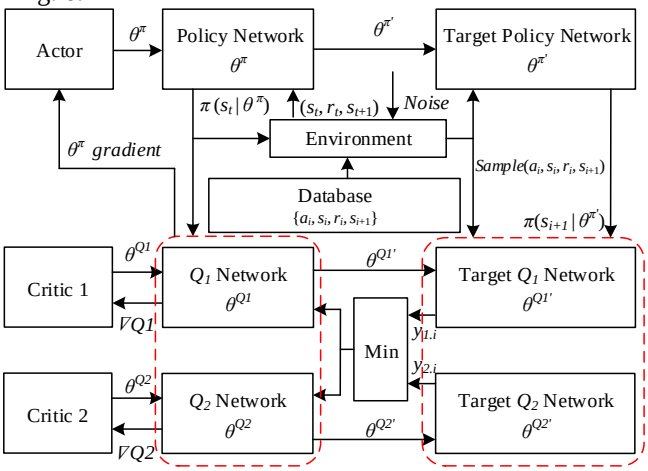


Fig. 6. Structure of TD3-DRL for ADRC.

There are three parts in the Agent of TD3 in this article, which can achieve the goal of convergence: 1) Actor, which can learn and build a policy network and select different *Action* according to current State. 2) Critic, which can evaluate the value of *Action*s to optimize the policy network. TD3 has two critics to mitigate the overestimation, and adopts delayed policy update to achieve policy update after fully minimizing the time difference error. 3) Noise, which is used to overcome the shortcomings of weak exploration of deterministic strategy algorithm, can broaden the scope of DRL, find the optimal decision and prevent it from falling into the local optimal. The deterministic policy gradient method is used to make the Critic converge and update the parameters of the network.

It is a routine operation in the DRL training field that Agent takes different a_t based on the current *State* and the added random noise ε

$$a_t = \pi\left(s_t \mid \theta^\pi\right) + \varepsilon \tag{14}$$

Noise ε added to the *Action* in the training process is generally Gaussian noise, and its standard deviation decreases with iteration. Reward r_t and next State s_{t+1} can be obtained when an Action has been done, and these data (s_t, a_t, r_t, s_{t+1}) will be stored in the database, and some data sets selected randomly from the database are used to train the TD3. $Q(s,a|\theta^Q)$ is used as the evaluation network of Critics. The self-optimization of the algorithm is achieved by updating θ^π and θ^Q. As shown in Fig.6, $\pi(s|\theta^\pi)$ and $Q(s,a|\theta^Q)$ are updated slowly by using $\pi'(s|\theta^{\pi'})$ and $Q'(s, \pi'(s|\theta^{\pi'})|\theta^{Q'})$, which are called the of target network of policy and Critic respectively. The update rate $\tau \leq 1$, and the update formulas are

$$\begin{cases} \theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \\ \theta^{\pi'} \leftarrow \tau\theta^\pi + (1-\tau)\theta^{\pi'} \end{cases} \tag{15}$$

The evaluation value y_i of the target Q network calculated from two Critics, and the loss L between the target network and the original one, can be obtained by Eq. (16). The smallest L is used as the optimization target to optimize Q network parameters θ^Q. The descending gradient according to the

random sampling datasets in Eq. (17) is used to update θ^{π} to maximize the evaluation value of Q.

$$\begin{cases} y_i = r_i + \gamma \min\limits_{j=1,2} Q'\left(s_{i+1}, \pi'\left(s_{i+1} \mid \theta^{\pi'}\right) \mid \theta^{Qj}\right) \\ L = \dfrac{1}{N}\sum\limits_{i}^{N}\left(\left(y_i - Q\left(s_i, a_i \mid \theta^{Qj}\right)\right)^2\right)_{j=1,2} \end{cases} \tag{16}$$

$$\nabla_{\theta^{\pi}} J(\theta^{\pi}) = \frac{1}{N}\sum_{i=1}^{N} \nabla_a Q\left(s_i, a \mid \theta^{Q}\right)|_{a=\pi(s_i|\theta^{\pi})} \nabla_{\theta^{\pi}} \pi(s_i \mid \theta^{\pi}) \tag{17}$$

The correction of Actions is penalized and rewarded as R_a to adjust the optimization by Eq. (18), and the final evaluation value is calculated by Eq. (19).

$$R_a = \frac{1}{n}\sum_{i=1}^{n}\left|R_a^i\right| \tag{18}$$

$$R = -\left(\alpha_{Obs} R_{Obs} + \alpha_a R_a\right) \tag{19}$$

where α_{Obs}, α_a are the corresponding weights of observation reward and adjust reward, respectively. R_{Obs} is the evaluation value of the training effect of the Agent according to the specific control system.

Algorithm 1 shows the policy pseudo code for training Agent through DRL-TD3. By setting the reward interface, Agent will modify and determine the original strategy network parameter value after each turn of training, and invoke the trained network through the online module to form a complete TD3-ADRC.

Algorithm 1 Training process of TD3

1:Initialize the parameters of Action, Critic, and target networks.
2:Initialize the sampling buffer of $\{s_t, a_t, r_t, s_{t+1}\}$.
3:**for** iteration = 1,2,… **do**
4: Employ the initial parameters to calculate the s_1 by state interference.
5: Select *Action* $a_t = \pi(s_t|\theta^{\pi})$
6: Simulate with a_t and obtain next *State* s_{t+1} and *Reward* r_t through state and reward interference.
7: Store the dataset $\{s_t, a_t, r_t, s_{t+1}\}$.
8: Sample a batch from the dataset after a certain rounds of interactions.
9: Calculate Eq. (16) by step. 8.
10: Update θ^{Qj} according to step. 9.
11: Update parameters of Actor θ^{π} by the deterministic policy gradient according to Eq. (17).
12. Update target networks according to Eq. (15).
end for

IV. Simulation of MSPO-DRL System

A. Optimization Setting of TD3-ADRC

Table I shows the parameters of the simulated motor model. For the sake of containing more kinds of disturbances and changes to test the effectiveness of the proposed method, the complex simulation environment is set up. In order to simulate the disturbance environment that MEA motors may encounter [7][8] and enhance the disturbance rejection ability of the proposed method, the start speed is set to 3000rpm and the start torque is set to $|2\sin(\pi t)|$ Nm. The periodic torque will change to a constant load of 5Nm at 3s, and the speed is changed to 5000rpm at 6s.

TABLE I
Main Parameters of PMSM

Symbol	Item	Value
R_s	phase resistance	0.0713Ω
L_q	q-axis inductance	605uH
L_d	d-axis inductance	519.5uH
Ψ_f	rotor flux	0.0201Wb
J	rotational inertia	$4.07\text{x}10^{-4}$kg.m^2
p_n	pole pairs	5

Every 100 times of TD3-ADRC calculation, DRL Agent will interact and learn with the environment once, sample the speed curve to update the State data set, and update the parameters of the DRL networks. In order to better improve the performance of the proposed method, the steady-state error of the control system, the disturbance rejection ability, and the error between the given torque and the actual torque, etc. are taken as the optimization objective. The value of observed Reward R_{obs} is designed as

$$R_{obs} = \frac{1}{n}\sum_{i=1}^{n}\left(\left|r_1 e_{os}/s_1 + r_2\,|e_l|/s_2 + r_3 e_{ts}/s_3 + r_4 t_{ss}/s_4 + r_5 t_{sl}/s_5\right|\right)^i \tag{20}$$

where e_{os} is the error of the observed state and reference state, e_l is the speed error of the sudden load transition, and e_{ts} is the error of the observed torque and reference torque respectively. t_{ss}, and t_{sl} are the increase time of speed and the recovery time of the disturbance load respectively, which could be calculated after the disturbances occur. $s_{1\text{-}5}$ are the standardization coefficients because of different dimensions between optimized objectives. $r_{1\text{-}5}$ are the weight coefficients which can be changed according to different demands of the application. The optimal results can be obtained after the value of R is converged to the smallest.

B. Convergence of the TD3-ADRC Method

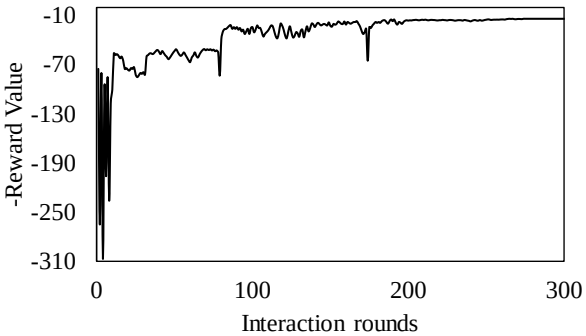


Fig. 7. Convergence curve of TD3-ADRC.

The DRL calculation is implemented in TensorFlow v2.10. and carried out by a computer configured with Inter(R) Core(TM) Inter(R) Core i7-10700KF, 32GB. The Agent learns synchronously with the simulation. In the procedure of training Agent, the noise disturbance is applied to the output of Action to avoid the Agent being trapped in the local optimization. The noise disturbance is set to change linearly, and gradually decreases with the round of interactions to demonstrate the real

effect of the algorithm. Figs. 7 and 8 show the convergence and disturbance of TD3-ADRC. The linear decreasing noise ε is added in the training process to broaden the scope of DRL, find the optimal decision and prevent it from falling into the local optimal. $\{s_t, a_t, r_t, s_{t+1}\}$ is stored and updated once for each interaction. The stored data set is used for learning every 100 calculations, and the TD3-ADRC network, that is the Actor network, is updated after each learning. It can be seen that the training converges around 240 rounds, that is, when the noise gradually becomes almost zero.

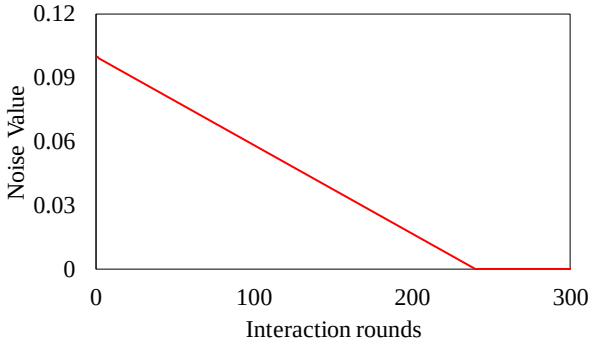


Fig. 8. Noise curve of TD3-ADRC.

C. *Comparison with ADRC*

In order to further demonstrate the performance of TD3-ADRC, Traditional ADRC is employed as a comparison. Except for the NLSEF part, the other parts and parameters of the two methods compared are identical. The control curve of TD3-ADRC in the learning process is also used for comparison and demonstration, which is shown as "TD3-ADRC learning". The proposed method after completing the training is shown as "TD3-ADRC". Fig. 9 shows the comparison results.

Since each step of TD3-ADRC *Action* in the learning process is exploratory, and the linearly decreasing Gaussian noise is added to the "*Action*" of each step which will cause minor deviations and fluctuations in DRL decisions, the curve of TD3-ADRC under learning jitters in a small range, and the jitter changes less and less with time and exploration rounds. Fig. 9(d) shows the choice between overshoot, response speed, and stability made by Agent in the learning process representatively. Although the blue curve has a faster response speed, the red curve has a more stable state and smaller overshoot, and could scores higher under the designed optimization objectives and constraints, which represents the learning result of Agent.

After TD3-ADRC has been sufficiently trained, it can be seen from Fig. 9 that TD3-ADRC has significantly improved the control performance compared with traditional ADRC. In terms of response speed, TD3-ADRC obtains better effect on both startup speed and recovery speed. The overshoot and interference of TD3-ADRC are also obviously less than those of ADRC. The data label in Fig.9 and the histogram in Fig. 10 show the comparison between the two methods more clearly.

When the control system is affected by periodic disturbance, it can be seen from Fig. 9(b) that though the traditional ADRC has good suppression effect against periodic disturbances, it can still be seen that the trend of the speed curve approximates to sinusoids while TD3-ADRC is hardly influenced by periodic disturbance and can maintain stable speed. It can be seen from Figs. 9 and 10 that the variety of TD3-ADRC is significantly smaller when it faces to the sudden increase of load, which is almost one third of that of ADRC. The feasibility and practical effect of the proposed method are demonstrated and verified.

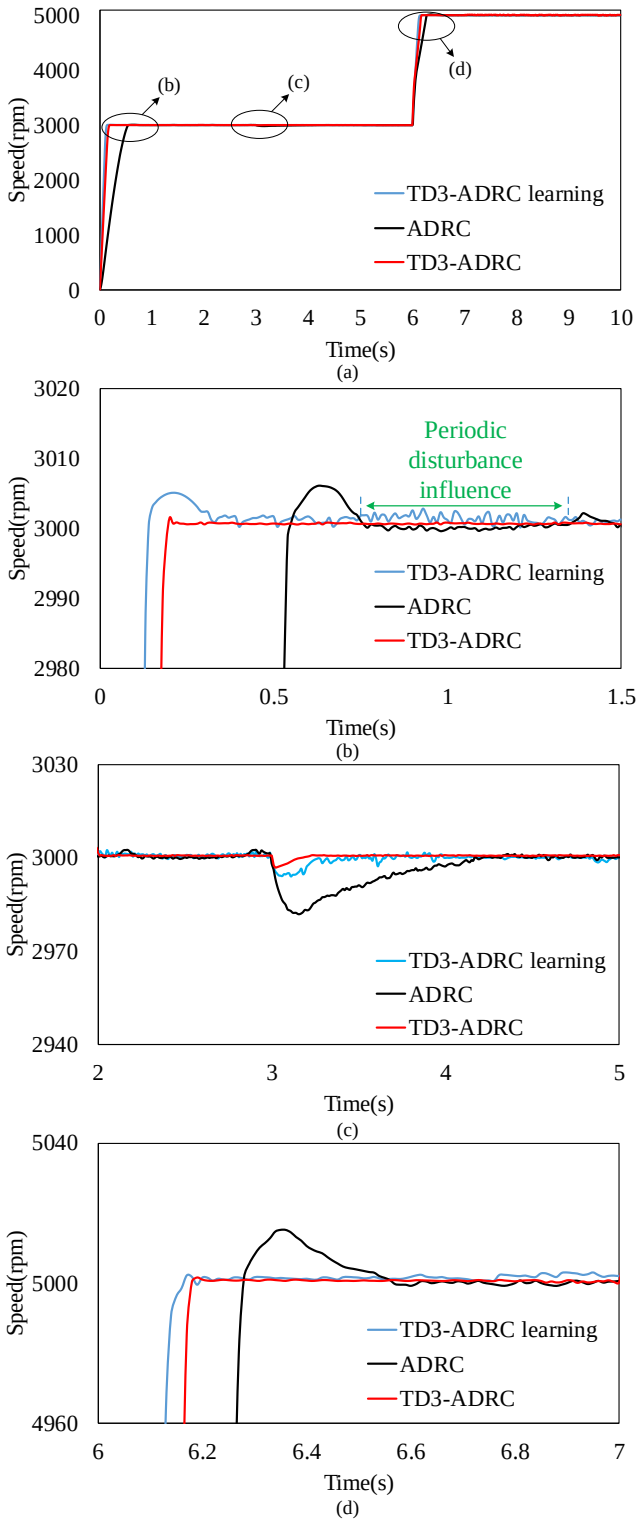


Fig. 9. Comparison of simulations under different external influences, (a) overview, (b)start-up state, (c)sudden load state, (d)sudden speed increase.

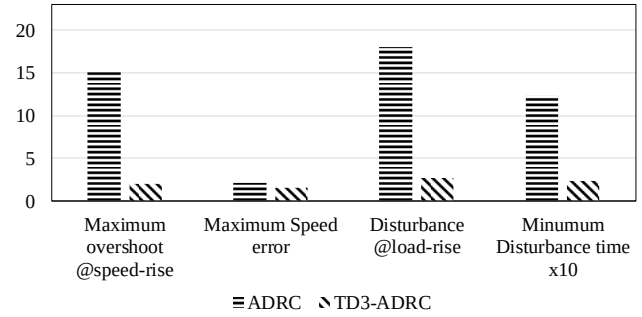


Fig. 10. Histogram of simulation results comparison.

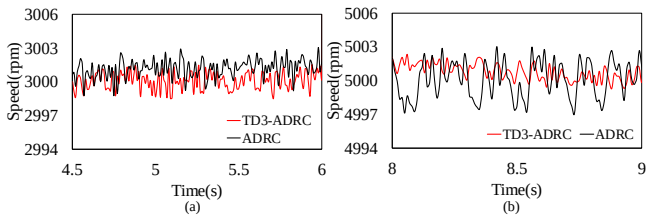


Fig. 11. Performances of internal sampling disturbance, (a)@3000rpm, (b)@5000rpm.

The sampling noise is added to the speed observation value to test the internal disturbance rejection ability of the proposed method. The mean value of the white noise of the added normal distribution is 0, the variance is 1, and the sampling time is 0.01s. It can be seen from Fig. 11 that ADRC and TD3-ADRC have similar performance at 3000 rpm, but the speed error of TD3-ADRC is smaller at 5000 rpm, which also means that TD3-ADRC has better suppression effect on internal disturbance and stable performance.

V. Experimental Verification

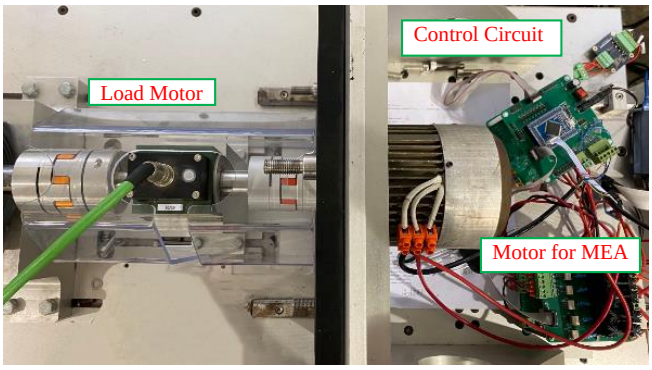
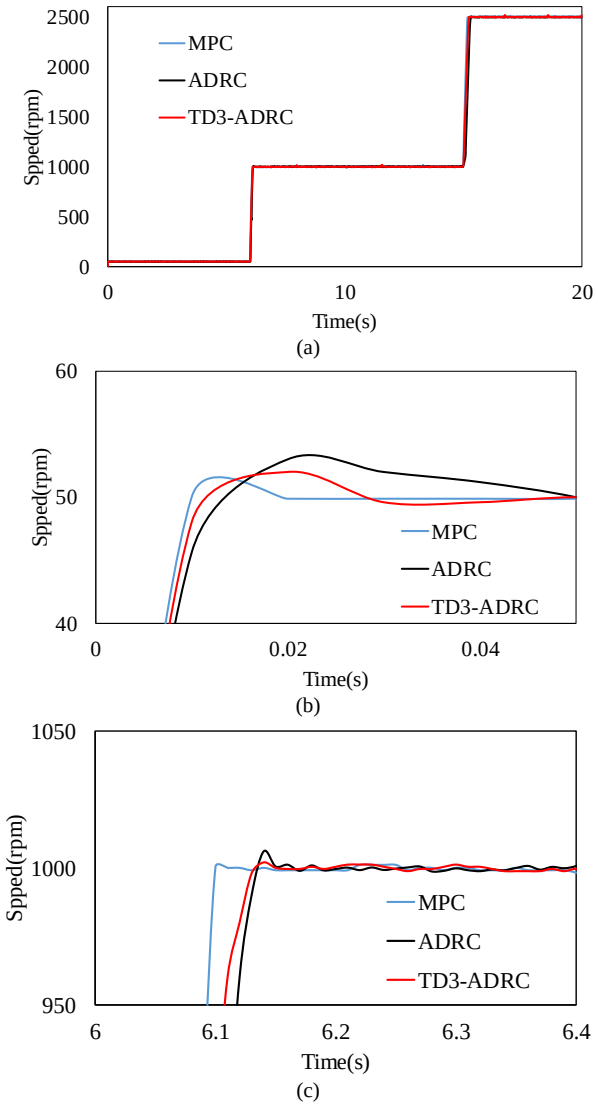


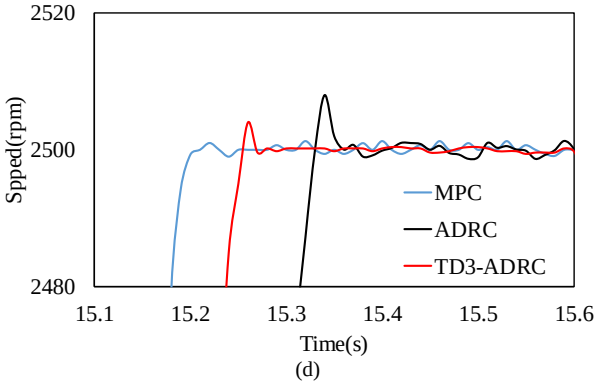
Fig. 12. Experimental platform.

In order to verify the control performance of theTD3-ADRC, the experimental platform is constructed as shown in Fig. 12. The experimental platform used in this article is MEA RDS loading experimental platform. All the control methods are implemented in micro controller unit (MCU) of STM32F407 with a clock frequency of 168 MHz, and the sampling period is 50 μs. The specific parameters of the motor are listed in Table I. This article replaces the control law with the Actor network of TD3 algorithm to interact with control system, and DRL uses data samples for training updating the Actor network after a certain number of interactions, which is independent of the control process and won't affect the calculation time. The execution time of TD3-ADRC which is the computational complexity is 46μs, and slightly shorter than 48μs of ADRC. Different experiments are carried out in order to test the control effect of the proposed control method in different application environments. MPC utilizes system discrete model and inherent discrete nature of motor inverter to forecast the future behavior of states and determines the future voltage vector according to optimization of an operating cost function at each sampling time[29][30]. In order to further verify the performance of the proposed method, MPC is also employed as an experimental method for comparison in addition to traditional ADRC.

A. *Speed experiments*

Different speed experiments are carried out to test the stability and fast response ability of the proposed method. The simulation results prove the excellent performance of TD3-ADRC. However, the DRL-based training method implies a training of the performance in response to sudden speed changes, and there is a generalization effect in the full speed range. Therefore, the speed is started from 50 rpm and gradually rises to 1000 rpm and 2500 rpm in the experiment to reflect the control effect of TD3-ADRC. Fig. 13 shows the comparison results of experiments.





(d)

Fig. 13. The speed response of three control methods under different cases, (a) overview, (b)start-up state to 50rpm, (c)50 to 1000rpm, (d)1000 to 2500rpm.

The parameters of ADRC are manually adjusted by trial-and-error method and experience method. Comparing TD3-ADRC with ADRC, the structure and parameters of other parts are identical except for the control law. It can be seen from Fig. 13 that compared with traditional ADRC, TD3-ADRC has smaller overshoot, smaller steady-state jitter and faster response speed. The overshoot of TD3-ADRC exists but very small when compared with MPC. MPC has the fastest response speed, and there is no overshoot at the start-up stage or when the speed is suddenly increased, but the speed error of steady-state is larger than TD3-ADRC.

B. Load experiments

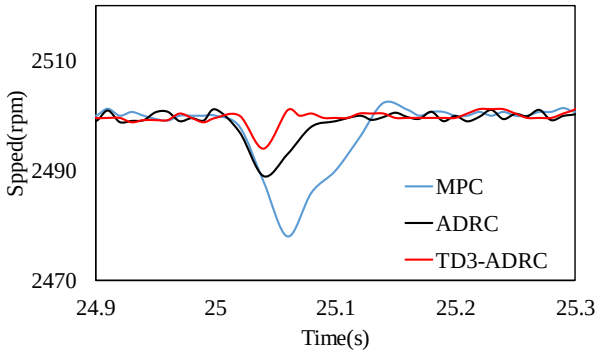


Fig. 14. Disturbances of three methods under sudden load rise of 1.5Nm.

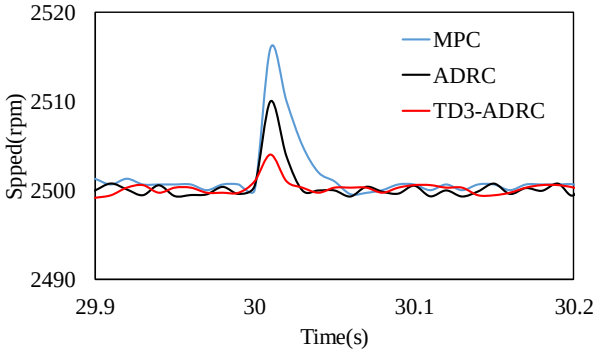


Fig. 15. Disturbances of three methods under sudden load reduce of 1Nm.

Load experiments are carried out to verify the external disturbance rejection ability of TD3-ADRC. The load test is performed when the speed of the motor is set to 2500 rpm. A sudden torque of 1.5Nm is applied to observe the disturbance rejection ability and recovery speed of different methods. Subsequently, the torque is changed to 0.5Nm. Figs. 14 and 15 show the comparisons of load experiments.

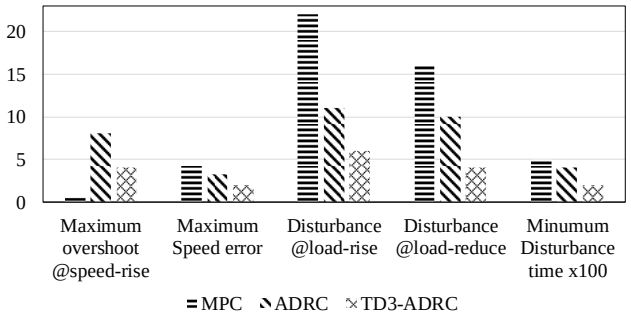


Fig. 16. Histogram comparisons of experimental results under different cases.

Classical MPC has a better dynamic response speed but is dependent on the accuracy of model parameters, and the disturbances of the motor will affect the performance of MPC. Therefore, in order to reject the excessive effect of disturbance, the 2-norm of input is added to the loss function of MPC in this article to limit the growth and drop the response speed. It can be seen from Figs. 14 and 15 that though MPC has achieved the best effect in speed experiment, it is worse than ADRC when the disturbance rejection ability is compared due to the differences of the controller principles. TD3-ADRC has the smallest disturbance error and the fastest recovery speed, followed by ADRC. The specific details of the speed and the load experiments can be more intuitively shown in Fig. 16.

It can be seen from Fig. 16 that since TD3-ADRC which uses neural network instead of control law needs calculation process, it is slightly worse than the optimal MPC in the aspect of response speed but the gap is very small, and both two methods are better than traditional ADRC. In the load experiment, TD3-ADRC has the best external disturbance rejection performance and is much better than the other two algorithms, which shows the superiority and effectiveness of the proposed method.

ADRC has strong disturbance rejection performance and is insensitive to parameter changes [15][21][31], which is one of the motivations for selecting ADRC for FW control system in the article. L_q, R_s, J, B are selected for parameter mismatch research to study whether the proposed method can be applied to suppress internal interference. The range for each parameter change is set from -80% to 80%. The effect of parameter mismatch is measured by the speed error with sudden load disturbance of 1.5Nm, and the comparison results of the three methods are shown in Fig. 17.

Since the absolute values of errors in different methods are different, Fig. 17 compares the absolute values of errors in different cases in the form of histograms, and compares the error percentages in the form of broken lines. Due to the FW in the control system, it is inevitable to be affected by parameter mismatch. It can be seen that MPC has the worst performance of parameter mismatch and the largest change in error percentage. When the parameter mismatch is less than 80%, the changes of MPC is small. But when the parameter mismatch exceeds 80%, MPC is obviously affected, especially the speed is unstable when facing the change of J in Fig. 17(c). In contrast, ADRC-series controllers have better internal

disturbance suppression effect, no matter in terms of error size or change trend. The rejection effect of ADRC is slightly weaker than that of TD3-ADRC. When TD3-ADRC is used for speed loop, it is insensitive to the changes of R and L. When J and B are mismatched, the trend of change is obvious, but both the trend and the absolute value of the error are still small. Generally, TD3-ADRC has the best performance in parameter mismatch, which also proves the internal disturbance rejection effect of the proposed method.

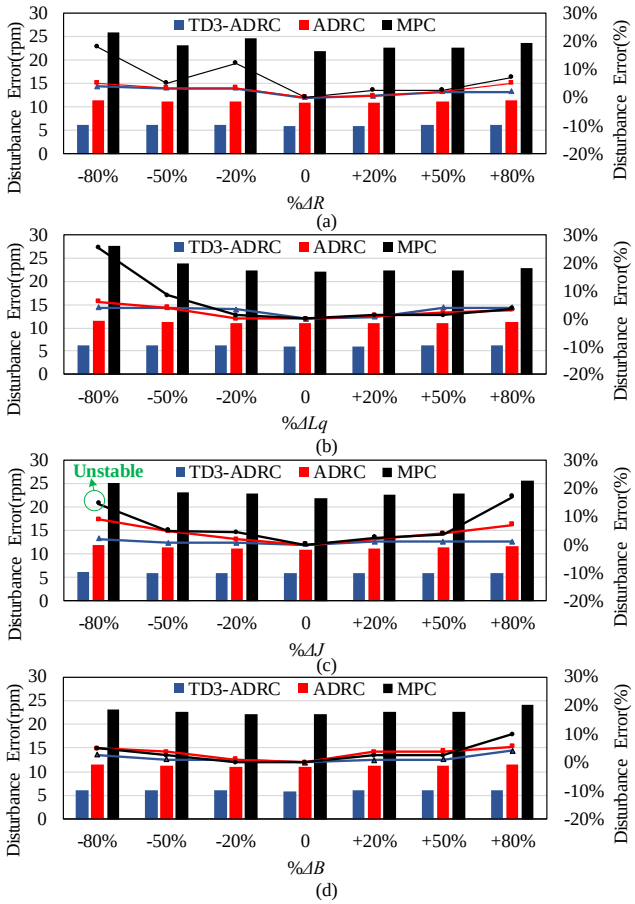


Fig. 17. Comparisons under different parameter mismatch cases of three methods. (a)R_s, (b)L_q, (c)J, (d)B.

VI. Conclusions

This article reviewed the FW and ADRC control of MEA motor and proposed a novel TD3-ADRC method. ADRC system was established based on the FW control used for MEA motor. MDP was integrated into the control system, and the interface module of DRL was thus designed. The proposed method used neural network to replace the traditional nonlinear state error feedback control law, and used DRL and TD3 algorithms to train the neural network. In addition to the traditional ADRC controller, the advanced MPC algorithm was also used for comparison. The control system was simulated and experimental platform were built. The results show that the trained TD3-ADRC has the same level of response speed as MPC, and more significant rejection effect in periodic disturbances and smaller speed fluctuations in sudden disturbances than MPC and ADRC, which verifies the effectiveness and superiority of the proposed method.

References

[1] E. Sayed et al., "Review of electric machines in more-/hybrid-/turbo-electric aircraft," *IEEE Trans. Transp. Electrif.*, vol. 7, no. 4, pp. 2976-3005, 2021.

[2] B. Sarlioglu, C. T. Morris, "More electric aircraft: review, challenges, and opportunities for commercial transport aircraft," *IEEE Trans. Transp. Electrif.*, vol.1, no.1, pp. 54-64, 2015.

[3] T. Zhao, S. Wu and S. Cui, "Multiphase PMSM with asymmetric windings for more electric aircraft," *IEEE Trans. Transp. Electrif.*, vol. 6, no. 4, pp. 1592-1602, 2020.

[4] Z. Song, C. Liu, K. Feng, H. Zhao and J. Yu, "Field prediction and validation of a slotless segmented-halbach permanent magnet synchronous machine for more electric aircraft," *IEEE Trans. Transp. Electrif.*, vol. 6, no. 4, pp. 1577-1591, 2020.

[5] M. A. A. Mohamed, M. Rashed and S. Bozhko, "Enhanced Flux Weakening Control of PMM-Based Starter-Generator System in More Electric Aircraft," in *Proc. 10th Int. Conf. Power Electron., Mach. Drives (PEMD)*, 2021, pp. 249-254.

[6] J. Wang, J. Wu, C. Gan and Q. Sun, "Comparative study of flux-weakening control methods for PMSM drive over wide speed range," in *Proc. 2016 19rd Int. Conf. Electr. Mach. Syst.* (*ICEMS*), 2016, pp. 1-6.

[7] W. Wang, "Design of flux weakening control system of PMSM based on the fuzzy self-tuning PID controller," in *Proc. Int. Conf. Consum. Electron., Commun. Netw.* (*CECNet*), Xianning, China, Apr. 2011, pp. 226-229.

[8] A. Wang, X. Jia and L. Zhang, "A new flux weakening control strategy considering voltage saturation for IPMSM drives," in *Chinese Control Conf.*, Hefei, July, 2012, pp. 4295-4299.

[9] Q. Zhang, M. Wang, H. Li and T. Cui, "Research on control characteristics of three-phase permanent magnet synchronous motor for electric aircraft propulsion," in *Chinese Control Conf. CCC*, Jul. 2020, pp. 2676-2681.

[10] J. X. J. Bannwarth, Z. J. Chen, K. A. Stol and B. A. MacDonald, "Disturbance accomodation control for wind rejection of a quadcopter," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, Jun. 2016, pp. 695-701

[11] J. Wang, D. Huang, S. Fang, Y. Wang and W. Xu, "Model predictive control for arc motors using extended state observer and iterative learning methods," *IEEE Trans. Energy Conver.*, vol. 37, no. 3, pp. 2217-2226, 2022.

[12] S. Wang, J. Na and Q. Chen, "Adaptive predefined performance sliding mode control of motor driving systems with disturbances," *IEEE Trans. Energy Conver.*, vol. 36, no. 3, pp. 1931-1939, 2021.

[13] B. Guo and F. Jin, "Sliding mode and active disturbance rejection control to stabilization of one-dimensional anti-stable wave equations subject to disturbance in boundary input," *IEEE Trans. Autom. Control*, vol. 58, no. 5, pp. 1269–1274, 2013.

[14] Z. Zhang, Z. Wang, X. Wei, Z. Liang, R. Kennel and J. Rodriguez, "Space-vector-optimized predictive control for dual three-phase PMSM with quick current response," *IEEE Trans. Power Electron.*, vol. 37, no. 4, pp. 4453-4462, 2022.

[15] J. Han, "From PID to active disturbance rejection control," *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 900-906, 2009.

[16] Y. Li, C. Zhang, J. Song, X. Li and B. Duan, "An Active disturbance rejection control strategy for a three-phase isolated matrix rectifier," *IEEE Trans. Transp. Electrif.*, vol. 8, no. 1, pp. 820-829, 2022.

[17] Y. Chen and Y. Zhou, "Active disturbance rejection and ripple suppression control strategy with model compensation of single-winding bearingless flux-switching permanent magnet motor," *IEEE Trans. Ind. Electron.*, vol. 69, no. 8, pp. 7708-7719, 2022.

[18] Y. Zuo, X. Ge, Y. Zheng, Y. Chen, H. Wang and A. T. Woldegiorgis, "An adaptive active disturbance rejection control strategy for speed-sensorless induction motor drives," *IEEE Trans. Transp. Electrif.*, vol. 8, no. 3, pp. 3336-3348, 2022.

[19] L. Zhu et al., "Nonlinear active disturbance rejection control strategy for permanent magnet synchronous motor drives," *IEEE Trans. Energy Conver.*, vol. 37, no. 3, pp. 2119-2129, 2022.

[20] Y. Xiao, S. Huang, G. Cao, H. Wang and J. Xu, "Active disturbance rejection position control of permanent magnet linear synchronous motors using feedback nonlinear state error," in *Proc. Int. Conf. Sens., Meas. Data Anal. era Artif. Intell.* (*ICSMD*), 2021, pp. 1-6.

[21] W. Ma, Y. Guan and B. Zhang, "Active disturbance rejection control

based control strategy for virtual synchronous generators," *IEEE Trans. Energy Conver.*, vol. 35, no. 4, pp. 1747-1761, 2020.

[22] Y. Bai et al., "High-gain nonlinear active disturbance rejection control strategy for traction permanent magnet motor drives," *IEEE Trans. Power Electron.*, vol. 37, no. 11, pp. 13135-13146, 2022.

[23] J. Hu, Q. Wang, Y. Ye and Y. Tang, "Toward online power system model identification: a deep reinforcement learning approach," *IEEE Trans. Power Syst.*, 2022, doi: 10.1109/TPWRS.2022.3180415(*in press*).

[24] Y. Tang et al., "Deep reinforcement learning-aided efficiency optimized dual active bridge converter for the distributed generation system," *IEEE Trans. Energy Conver.*, vol. 37, no. 2, pp. 1251-1262, 2022.

[25] S. Park et al., "A deep reinforcement learning framework for fast charging of Li-ion batteries," *IEEE Trans. Transp. Electrif.*, vol. 8, no. 2, pp. 2770-2784, 2022.

[26] M. Nicola and C. I. Nicola, "Tuning of PI speed controller for PMSM control system using computational intelligence," in *2021 21st Int. Symp. Power Electron.* (*Ee*), 2021, pp. 1-6.

[27] L. Yan, X. Chen, Y. Chen and J. Wen, "A cooperative charging control strategy for electric vehicles based on multi-agent deep reinforcement learning," *IEEE Trans. Ind. Inf.*, 2022, doi: 10.1109/TII.2022.3152218 (*in press*).

[28] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn.* (*ICML*), Stockholm, Sweden, 2018, pp. 1587–1596.

[29] D. Ye, J. Li, R. Qu and H. Lu, "Model predictive control of dual three-phase interior permanent magnet synchronous generator for more electrical aircraft," in *Proc. 2018 21st Int. Conf. Electr. Mach. Syst.* (*ICEMS*), 2018, pp. 1258-1261.

[30] Y. Luo and C. Liu, "Multi-vector-based model predictive torque control for a six-phase PMSM motor with fixed switching frequency," *IEEE Trans. Energy Conver.*, vol. 34, no. 3, pp. 1369-1379, Sept. 2019.

[31] V. R. Teja, C. Chakraborty and B. C. Pal, "Disturbance rejection analysis and FPGA-based implementation of a second-order sliding mode controller fed induction motor drive," *IEEE Trans. Energy Conver.*, vol. 33, no. 3, pp. 1453-1462, 2018.